Available online at www.sciencedirect.com

SCIENCE DIRECT®

Discrete Applied Mathematics 140 (2004) 169–183

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Searching in trees☆

Frank Recker[a,1]

[a] *Department of Mathematics, University of Hagen, D-58084 Hagen, Germany*

Received 4 December 2001; received in revised form 23 May 2003; accepted 26 May 2003

Abstract

In (Discrete Math. 17 (1977)181) Rivest introduced the search complexity of binary trees and proved that among all binary trees with a fixed search complexity the smallest ones are the so-called Fibonacci trees. This result is extended for q-trees. The structure of the smallest q-trees is again Fibonacci-like but more complicated than in the binary case. In addition an upper bound for the asymptotic growth of these trees is given.

Keywords: Combinatorial search; q-Tree; Fibonacci tree

1. Introduction and Results

In [3] Rivest analyzed the search complexity of (rooted) binary trees. A *rooted tree* is a tuple (T,r) with a tree T and a node r in T, called the *root* of T. If the root is not important we will write T instead of (T,r). We will write $x \in T$ if x is a node of T. For all nodes $x \in T$ the subtree T_x of T consists of x and all transitive children of x.

The *combinatorial search problem* can be described as a game between two Players A and B. A wants to find a leaf in the tree. In every round A chooses a node x and asks B. B answers whether the unknown leaf is in T_x or not. B is usually called the oracle. The only constraint to B's answers is that there must always be at least one leaf which is consistent with all answers given so far. There is no need for B to choose a fixed leaf at the beginning of the game. We refer to [1] or [2] for a complete introduction into combinatorial search.

☆ Parts of this work appeared in [4].

[1] Partially supported by the "Deutsche Forschungsgemeinschaft" (We 1265/2–1).

E-mail address: frank.recker@fernuni-hagen.de (F. Recker).

0166-218X/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2003.05.001

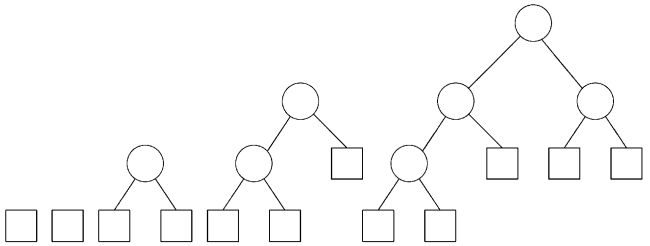

Fig. 1. The Trees $F_1,\ldots,F_5$.

Given a questioning strategy for A and an answering strategy for B the game will eventually end after $L(A,B)$ rounds. In [1] the following minimax-theorem is proven:

$$\min_A \max_B L(A,B) = \max_B \min_A L(A,B), \tag{1}$$

where the minima and maxima are taken over all possible strategies. The main insight is that this number depends only on the tree T. It is called the *search complexity of* T, denoted by $C(T)$.

Rivest solved the following problem: Let $r \in \mathbb{N}$, $r \geqslant 1$. What is the smallest size (measured in the number of leafs) of a binary tree with search complexity at least r?

The answer can be given with the help of Fibonacci trees. F_1 and F_2 consist of just one leaf. For $i \geqslant 3$, the tree F_i is constructed by taking a new root and connecting it with the roots of F_{i-2} and F_{i-1}. Fig. 1 shows the first five Fibonacci trees.

The Fibonacci tree F_i has exactly f_i leafs where f_i is the ith Fibonacci Number.

The solution to the problem mentioned above is given by Rivest in [3]. Let $r \in \mathbb{N}$, $r \geqslant 1$. The Fibonacci tree F_{r+2} has search complexity r. Any binary tree T with $C(T) \geqslant r$ has at least f_{r+2} leafs. The bound is therefore sharp.

Given the usual formula for the Fibonacci numbers the minimal number of leafs for r questions is asymptotically $(1/\sqrt{5})\theta^{r+2}$ where $\theta = (1+\sqrt{5})/2 \approx 1.618034$ is the golden ratio.

In this paper the class of trees is extended to q-trees. A q-tree is a rooted tree in which every node has at most q children. Therefore a binary tree is a 2-tree. A complete introduction into q-trees can be found in [1].

For q-trees equation (1) is also true (cf. [1] for a general minimax-theorem) and the value is also called the search complexity $C(T)$. Rivest's question for q-trees is: *Let* $q,r \in \mathbb{N}$, $q \geqslant 2$ *and* $r \geqslant 1$. *What is the minimal number of leafs of a* q*-tree with search complexity at least* r?

It turns out that stating the result for general q is quite complicated. For a q-tree T we write $|T|$ for the number of leafs of T. By $g(T)$ we denote the maximum of $|T_x|$ for all proper subtrees T_x of T and by $s(T)$ the number of children of the root of T. The Fibonacci trees have the values $|F_i| = f_i$ (F_i has f_i leafs), $g(F_i) = f_{i-1}$ (the biggest proper subtree T_x of F_i is F_{i-1}) and $s(F_i) = 2$ (the root of F_i has two children).

Let $\leqslant$ be the canonical partial ordering on $\mathbb{N}^3$ given by

$$(x, y, z) \leqslant (x', y', z') \;\Leftrightarrow\; x \leqslant x', \quad y \leqslant y', \; z \leqslant z'.$$

An element x in $A \subseteq \mathbb{N}^3$ is minimal if there is no $y \in A$ with $y < x$ (i.e. $y \leqslant x$ and $y \neq x$). We write $\min A$ for the set of all minimal elements of a nonempty subset $A \in \mathbb{N}^3$.

For the formulation of the main result we need the algorithm CALCMIN. For $q, R \in \mathbb{N}$, $q \geqslant 2$, $R \geqslant 1$ CALCMIN is given by

$m_{q,1} := \{(2,1,2)\}$
For $r = 2, \ldots, R$ **do**
  $m'_{q,r} := \emptyset$
  For all $(n,a,b) \in m_{q,r-1}$ **do**
    $m'_{q,r} := m'_{q,r} \cup \{(n+a, n, 2)\}$
    If $b < q$ **then**
      $m'_{q,r} := m'_{q,r} \cup \{(n+a, a, b+1)\}$
    Endif
  Endfor
  $m_{q,r} := \min m'_{q,r}$
Endfor

Theorem 1. *Let $q, R \in \mathbb{N}$, $q \geqslant 2$ and $R \geqslant 1$. Then for every $r = 1, \ldots, R$ CALCMIN calculates a set $m_{q,r}$ of triples of natural numbers.*

For every $(n,a,b) \in m_{q,r}$ exists a q-tree T with $|T| = n$, $g(T) = a$, $s(T) = b$ and $C(T) = r$.

For every q-tree T with $C(T) \geqslant r$ exists $(n,a,b) \in m_{q,r}$ with $n \leqslant |T|$, $a \leqslant g(T)$ and $b \leqslant s(T)$.

Theorem 1 can be restated using the canonical partial ordering of $\mathbb{N}^3$.

Corollary 2. *Let $q, r \in \mathbb{N}$, $q \geqslant 2$, $r \geqslant 1$ and $m_{q,r}$ as in Theorem 1. Then*

$$m_{q,r} = \min\{(|T|, g(T), s(T)) | T \text{ is a } q-\text{tree}, \; C(T) \geqslant r\}.$$

This enables us to calculate the minimal number of leafs of all q-trees with a given search complexity. Let T_1 be a q-tree with search complexity at least r. From Theorem 1 follows that either $(|T_1|, g(T_1), s(T_1)) \in m_{q,r}$ or there is a tree T_2 with $(|T_2|, g(T_2), s(T_2)) < (|T_1|, g(T_1), s(T_1))$ and $(|T_2|, g(T_2), s(T_2)) \in m_{q,r}$. This shows the existence of a tree T with $(|T|, g(T), s(T)) \in m_{q,r}$ and minimal number of leafs of all q-trees with search complexity at least r.

Corollary 3. *Let $q \geqslant 2$, $r \geqslant 1$. Then the minimal number of leafs of all q-trees with search complexity at least r is*

$$\mu_{q,r} := \min\{n \in \mathbb{N} \mid \exists (n,a,b) \in m_{q,r}\}.$$

Table 1
An example of the calculation of CALCMIN

r	$m'_{3,r}$	$m_{3,r}$	$\mu_{3,r}$
1		(2,1,2)	2
2	(3,2,2),(3,1,3)	(3,2,2),(3,1,3)	3
3	(5,3,2),(5,2,3),(4,3,2)	(5,2,3),(4,3,2)	4
4	(7,5,2),(7,4,2),(7,3,3)	(7,4,2),(7,3,3)	7
5	(11,7,2),(11,4,3),(10,7,2)	(11,4,3),(10,7,2)	10

Table 2
The values of $\mu_{q,r}$

	r												
q	1	2	3	4	5	6	7	8	9	10	11	12	13
2	2	3	5	8	13	21	34	55	89	144	233	377	610
3	2	3	4	7	10	15	24	37	56	89	137	209	330
4	2	3	4	5	9	13	17	24	40	56	81	115	185
5	2	3	4	5	6	11	16	21	26	35	60	85	110
6	2	3	4	5	6	7	13	19	25	31	37	48	84
7	2	3	4	5	6	7	8	15	22	29	36	43	50
8	2	3	4	5	6	7	8	9	17	25	33	41	49
9	2	3	4	5	6	7	8	9	10	19	28	37	46
10	2	3	4	5	6	7	8	9	10	11	21	31	41

We remark that a q-tree T with $C(T) \geqslant r$ and $|T| = \mu_{q,r}$ fulfills $C(T) = r$: Assume to the contrary that $C(T) > r$. We delete one leaf x of T and get the q-tree T'. Every search algorithm A' for T' can be extended to a search algorithm A for T by adding the question T_x. This shows $C(T') \geqslant C(T) - 1 \geqslant r$. Hence $\mu_{q,r} \leqslant |T'| < |T|$ which is a contradiction to $|T| = \mu_{q,r}$.

Table 1 shows an example for CALCMIN for the case $q = 3$. For $r = 1$ there is no set $m'_{3,1}$ by definition. For $r \geqslant 2$ the set $m_{3,r}$ contains the minimal triples of $m'_{3,r}$. By Corollary 3 the value of $\mu_{3,r}$ is the minimal number of leafs of all 3-trees with search complexity at least r.

In this way we can calculate the minimal number of leaves of a q-tree with search complexity r. Table 2 gives some values of $\mu_{q,r}$. As far as we know for $q \geqslant 3$ there isn't any linear recurrence equation for the series $(\mu_{q,r})_{r\in\mathbb{N}}$. The values are obtained as the "minimal first component" of the sets in $m_{q,r}$ (cf. Corollary 3).

As in [3] we analyzed the asymptotic growth of the value $\mu_{q,r}$. A partial solution is given by:

Theorem 4. *For all $q \geqslant 2$ the series $\left(\sqrt[r]{\mu_{q,r}}\right)_{r\in\mathbb{N}}$ converges and*

$$\lim_{r\to\infty} \sqrt[r]{\mu_{q,r}} \leqslant \sqrt{\frac{q+1+\sqrt{q^2+2q-3}}{2}}.$$

For $q = 2$ this bound is equal to ϕ and therefore equal to the limit. Extensive computer based calculations support the belief that this is true for all $q \geqslant 3$, which gives raise to:

Conjecture 5. *For all* $q \in \mathbb{N}$, $q \geqslant 3$

$$\lim_{r\to\infty} \sqrt[r]{\mu_{q,r}} = \sqrt{\frac{q+1+\sqrt{q^2+2q-3}}{2}}.$$

In Section 2 we will prove an upper bound that yields the first part of Theorem 1. The lower bound will be shown in Section 3 and the bound for the asymptotic value will be shown in Section 4. Further remarks on the conjecture will be given in Section 5.

2. The upper bound

In this section we will prove the first part of Theorem 1, i.e. the existence of a q-tree T with the desired properties. To this end, we will introduce a set of trees which can be seen as a generalization of the Fibonacci trees to the nonbinary case.

For all $q, r \in \mathbb{N}$ with $q \geqslant 2$ and $r \geqslant 1$ let

$$\mathscr{T}_{q,r} := \{T \mid T \text{ is a } q-\text{tree}, \ C(T) \geqslant r\}.$$

The search complexity is monotone: If a tree T is a subtree of a tree S then $C(T) \leqslant C(S)$.

Proof (Sketch). An optimal search algorithm for the tree S can be applied to the tree T. As a consequence if $T \in \mathscr{T}_{q,r}$ and T is a subtree of S then also $S \in \mathscr{T}_{q,r}$.

Let $T_1, T_2, \ldots, T_n$ be trees with roots $r_1, \ldots, r_n$. The tree $(T_1, T_2, \ldots, T_n)$ is constructed by connecting a new root r to $r_1, r_2, \ldots, r_n$, such that r is the root of the tree $(T_1, T_2, \ldots, T_n)$. Then Fibonacci trees have the form

$$F_i = (F_{i-1}, F_{i-2}) \quad \text{for all } i \geqslant 3. \tag{2}$$

Obviously F_{i-2} is a subtree of F_{i-1} for all $i \geqslant 3$. Moreover F_{i-2} is the biggest proper subtree of F_{i-1}. Therefore $|F_{i-2}| = g(F_{i-1})$. This is the property which allows a generalization to q-trees. □

Definition 6. Let $T = (T_1, T_2, \ldots, T_n)$ be a tree. Then we define

$$Y(T) = \{(T, T_i) \mid i \in \{1, \ldots, n\}, \ |T_i| = g(T)\}$$

and

$$N(T) = \{(T_i, T_1, T_2, \ldots, T_n) \mid i \in \{1, \ldots, n\}, |T_i| = g(T)\}.$$

Eq. (2) becomes

$$F_i \in Y(F_{i-1}) \quad \text{for all } i \geqslant 3.$$

From now on we assume that all trees T with at least two leafs can be written as $T=(T_1,\ldots,T_n)$ for some $n \geqslant 2$ with $|T_1| \geqslant |T_2| \geqslant \cdots \geqslant |T_n|$. For $n=1$ the root of T is connected to only one node x and the tree T has the same search complexity as the tree T_x. A reordering of the subtrees does not change the search complexity either so the assumption can always be fulfilled.

We can now define the q-Fibonacci trees for $q \geqslant 2$. Different from the case $q=2$ these are not unique anymore. This is significant as will be seen at the end of Section 3.

Definition 7. Let $q \in \mathbb{N}$, $q \geqslant 2$. The set $\mathscr{F}_{q,1}$ consists of one tree with a root and two leafs. For all $r \in \mathbb{N}$, $r \geqslant 2$ let

$$\mathscr{F}_{q,r+1} := \left(\bigcup_{T \in \mathscr{F}_{q,r}} Y(T) \right) \cup \left(\bigcup_{T \in \mathscr{F}_{q,r} \wedge s(T) < q} N(T) \right).$$

A tree $T \in \mathscr{F}_{q,r}$ is called a (q,r)-Fibonacci tree.

Observe that for all $r \geqslant 1$, the set $\mathscr{F}_{2,r}$ has just one element which is the Fibonacci tree F_{r+2}. For general q all trees in $\mathscr{F}_{q,r}$ are q-trees since $N(T)$ is only added for trees T with $s(T) < q$.

Note the similarity between CALCMIN and Definition 7.

Lemma 8. *For all* $q,r \in \mathbb{N}$, $q \geqslant 2$, $r \geqslant 1$

$$m_{q,r} = \min\{(|T|, g(T), s(T)) \mid T \in \mathscr{F}_{q,r}\}.$$

Proof (Basis). $r=1$. $m_{q,1}=(2,1,2)$ and $\mathscr{F}_{q,1}$ has exactly one element T with $(|T|, g(T), s(T)) = (2,1,2)$.

Step: $r-1 \to r$. First we show that for all $(n,a,b) \in m_{q,r}$ there is a tree $T \in \mathscr{F}_{q,r}$ with $(|T|, g(T), s(T)) = (n,a,b)$. Then we show that for all trees $T \in \mathscr{F}_{q,r}$ there is a triple $(n,a,b) \in m_{q,r}$ with $(n,a,b) \leqslant (|T|, g(T), s(T))$. Together this proves the lemma.

Let $(n,a,b) \in m_{q,r}$. By construction there has to be $(n',a',b') \in m_{q,r-1}$ with $(n,a,b) = (n'+a, n', 2)$ or $(n,a,b) = (n'+a', a', b'+1)$ and $b' < q$. By induction there is a tree $S \in \mathscr{F}_{q,r-1}$ with $(|S|, g(S), s(S)) = (n',a',b')$. Choose $T \in Y(S)$ for the first case and $T \in N(S)$ for the second case. This gives $T \in \mathscr{F}_{q,r}$ and $(|T|, g(T), s(T)) = (n,a,b)$ in both cases.

Now let $T \in \mathscr{F}_{q,r}$. By definition $T \in Y(S)$ or $T \in N(S)$ with $S \in \mathscr{F}_{q,r-1}$. In the second case $s(S) < q$ since otherwise T would not be in $\mathscr{F}_{q,r}$. By induction there is $(n',a',b') \in m_{q,r-1}$ with $(n',a',b') \leqslant (|S|, g(S), s(S))$. Now let (n,a,b) be $(n'+a', n', 2)$ for the first case and $(n'+a', a', b'+1)$ for the second case. In both cases we have $(n,a,b) \in m_{q,r}$ and $(n,a,b) \leqslant (|T|, g(T), s(T))$ by the definition of Y and N. $\square$

Lemma 9. *Let $q,r\in\mathbb{N}$, $q\geqslant 2$ and $r\geqslant 1$. Then all $T=(T_1,\ldots,T_l)\in\mathscr{F}_{q,r}$ have the following property*:

$$T_1=T_2=\cdots=T_{l-1}$$

and T_l is either equal to T_1 or equal to a largest subtree of T_1. Furthermore, all largest subtrees of T are isomorphic.

Proof. For $r=1$ the unique tree in $\mathscr{F}_{q,r}$ fulfills the claim. For $r>1$ the claim follows easily by induction and the definition of $Y(T)$ and $N(T)$. □

Let T be a tree and $x\in T$. If a search algorithm asks for the tree T_x then an oracle can answer "Yes" or "No". In the first case the total number of questions needed to find the leaf will be at least $1+C(T_x)$. In the second case let $T-T_x$ be the tree which has exactly those leafs of T that are not in T_x. Then $1+C(T-T_x)$ questions will be necessary. Since x can be chosen arbitrary we conclude that

$$C(T)=1+\min_{x\in T}\max\{C(T_x),C(T-T_x)\}. \tag{3}$$

Lemma 10. *Let $q,l\in\mathbb{N}$, $q\geqslant 2$, $l\geqslant 3$ and $T_1,\ldots,T_l$ q-trees. Then the trees $(T_1,(T_2,T_3,\ldots,T_l))$, $(T_2,(T_1,T_3,\ldots,T_l))$ and $(T_1,T_2,T_3,\ldots,T_l)$ all have the same search complexity.*

Proof. As an example we show

$$C((T_1,(T_2,T_3,\ldots,T_l)))\leqslant C((T_2,(T_1,T_3,\ldots,T_l))).$$

Let $A:=(T_1,(T_2,T_3,\ldots,T_l))$, $B:=(T_2,(T_1,T_3,\ldots,T_l))$ and $k:=C(B)$.

Basis: $k=0$. B can have at most one leaf. Therefore A has at most one leaf and $C(A)=0$.

Step: $k\to k+1$. Let the first question in A be the tree T_x. We may assume w.l.o.g. that x is not the root of $(T_1,T_3,\ldots,T_l)$ since otherwise the algorithm can ask for T_2 instead. Let y_x be defined as follows: If x is the root of B then y_x is the root of A. If x is a node in the subtree T_i of B then y_x is the same node in the copy of T_i in A.

The trees A_{y_x} and B_x are equal or fulfill the condition of Lemma 10. Because of Eq. (3) there must a node $x\in B$ with $C(B_x)\leqslant k$ and $C(B-B_x)\leqslant k$. By induction $C(A_{y_x})\leqslant k$ and $C(A-A_{y_x})\leqslant k$. Therefore $C(A)\leqslant k+1=C(B)$ by Eq. (3). □

Fig. 2 illustrates Lemma 10.

Lemma 11. *Let $q\in\mathbb{N}$, $q\geqslant 2$. Then for all $r\in\mathbb{N}$, $r\geqslant 1$ and all $T\in\mathscr{F}_{q,r}$*

$$C(T)=r.$$

Proof. The tree in $\mathscr{F}_{q,1}$ has search complexity 1.

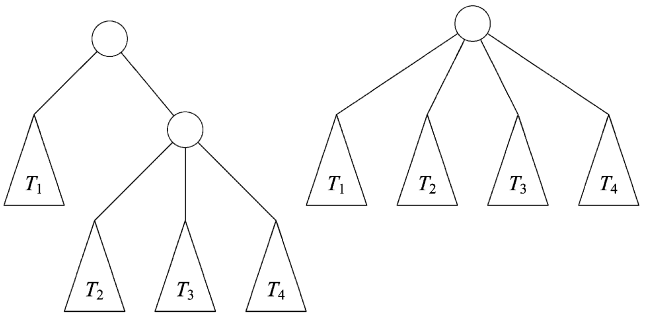

Fig. 2. Two trees with the same search complexity.

Let $r > 1$. A tree $T=(T_1,\ldots,T_n)\in\mathscr{F}_{q,r}$ must be in $Y(S)$ or in $N(S)$ for $S\in\mathscr{F}_{q,r-1}$. We show that $C(T)\leqslant C(S)+1$ and $C(T)\geqslant C(S)+1$. Since by induction $C(S)=r-1$ this completes the proof.

Let $S=(S_1,\ldots,S_l)$.

Case 1: $T\in N(S)$. The definition of N and Lemma 9 give $T=(S_1,S_1,\ldots,S_l)$. Let the algorithm ask for the tree S_1. If the answer is "No" then the leaf must be in S and therefore $C(S)$ further questions are enough to find it. If the answer is "Yes" then the leaf is in S_1. Since S_1 is a subtree of S, $C(S_1)$ questions are necessary with $C(S_1)\leqslant C(S)$. This implies $C(T)\leqslant C(S)+1$.

By Lemma 9 the trees $S_1,\ldots,S_{l-1}$ are isomorphic and the tree S_l is isomorphic to a subtree of S_1. Whatever tree is asked for by the search algorithm the oracle can answer "No" so that the remaining tree has S as a subtree. This proves that $C(T)\geqslant C(S)+1$.

Case 2: $T\in Y(S)$. Here we have $T=(S_1,(S_1,\ldots,S_l))$. By Lemma 10 we have also $C(T)=C(S_1,S_1,\ldots,S_l)$. But $C(S_1,S_1,\ldots,S_l)=r$ was already shown in Case 1. □

We can conclude from the proof that in an optimal game (i.e. A and B play optimal) the first answer of B is "Yes" for a tree in $Y(T)$ and "No" for a tree in $N(T)$. This is the reason for the operator names.

Proof of the first part of Theorem 1. Let $(n,a,b)\in m_{q,r}$. From Lemma 8 follows the existence of q-tree $T\in\mathscr{F}_{q,r}$ with $(n,a,b)=(|T|,g(T),s(T))$. Lemma 11 yields $C(T)=r$. □

3. The lower bound

We will now prove the second part of Theorem 1. Moreover we will show that the Fibonacci trees are examples of minimal trees for a given search complexity.

In the following we will write $S \leqslant T$ for $(|S|, g(S), s(S)) \leqslant (|T|, g(T), s(T))$ and $S < T$ for $(|S|, g(S), s(S)) < (|T|, g(T), s(T))$.

Lemma 12. *For all $q, r \in \mathbb{N}$ with $q \geqslant 2$ and $r \geqslant 1$ and all q-trees $T \in \mathscr{T}_{q,r}$ there is a tree $S \in \mathscr{F}_{q,r}$ with $S \leqslant T$.*

Since $S \leqslant T$ implies $|S| \leqslant |T|$ this Lemma already shows the claim at the beginning of this section about minimal examples for a given search complexity.

Proof (Basis $r = 1$). We have $(|T|, g(T), s(T)) \geqslant (2, 1, 2)$ since T has at least two leaves. For the tree $S \in \mathscr{F}_{q,1}$ we have $(|S|, g(S), s(S)) = (2, 1, 2)$.

Step: $r \to r + 1$. Assume that T is a q-tree with $C(T) \geqslant r + 1$ and that there is no tree $S \in \mathscr{F}_{q,r+1}$ with $S \leqslant T$. Let T be a minimal counterexample (there is no counterexample U with $U < T$). T can be written as $(T_1, T_2, \ldots, T_l)$. W.l.o.g. assume that $|T_1| \geqslant |T_2| \geqslant \cdots \geqslant |T_l|$ and $l \geqslant 2$.

Claim 1. For the tree $T' := (T_2, \ldots, T_l)$ we have $C(T') < r$.

Otherwise, by induction there is a tree $S' \in \mathscr{F}_{q,r}$ with $S' \leqslant T'$. Since $s(T') = s(T) - 1$ it follows that $s(S') < q$. Therefore $S'' \in N(S')$ is a tree in $\mathscr{F}_{q,r+1}$ with $|S''| = |S'| + |T_2| \leqslant |S'| + |T_1| = |T|$, $g(S'') \leqslant |T_2| \leqslant |T_1| = g(T)$ and $s(S'') = s(S') + 1 = s(T)$. This contradicts the assumption that no tree S'' in $\mathscr{F}_{q,r+1}$ with $S'' \leqslant T$ exists.

Claim 2. We have $C(T_1) \geqslant r$.

Otherwise, by Eq. (3) and Claim 1 we have $C(T) \leqslant r$ which contradicts $C(T) \geqslant r + 1$.

Claim 2 and the induction hypothesis yield the existence of a tree $S_1 \in \mathscr{F}_{q,r}$ with $S_1 \leqslant T_1$. Choose $S \in Y(S_1) \subset \mathscr{F}_{q,r+1}$. By definition of $Y(S_1)$ we have $s(S) = 2$ and $|S| \leqslant 2|S_1|$.

Claim 3. $2|T_1| > |T|$.

Assume $2|T_1| \leqslant |T|$. We have $g(T) = |T_1|$, $S_1 \leqslant T_1$ and $|S| \leqslant 2|S_1|$. Therefore

$$(|T|, g(T), s(T)) \geqslant (2|T_1|, |T_1|, s(T)) \geqslant (2|S_1|, |S_1|, 2) \geqslant (|S|, g(S), s(S)).$$

This means $S \leqslant T$ which contradicts the assumption about T.

Claim 4. $s(T) = 2$.

Assume $S(T) > 2$. Apply Lemma 10 on the tree T and its subtree T_1 and get a tree $U = (T_1, (T_2, \ldots, T_l))$ with $C(U) = C(T)$, $|U| = |T|$ and $s(U) = 2$. Because of $2|T_1| > |T| = |U|$ (Claim 3) T_1 is the largest subtree of U and therefore $g(U) = |T_1|$

$=g(T)$. Therefore $U < T$ and U is a counterexample for the lemma. This contradicts the choice of T as a minimal counterexample.

Claim 5. $g(T_1) \leqslant |T| - |T_1|$.

Assume the claim is not true. Then Lemma 10 allows to build a smaller counterexample U as follows: Let $T = (T_1, T_2)$ (Claim 4). We assume w.l.o.g. that $T_1 = (T_{1,1}, T_{1,2}, \ldots, T_{1,p})$ with $p \geqslant 2$ and

$$|T_{1,1}| \geqslant |T_{1,2}| \geqslant \cdots \geqslant |T_{1,p}|.$$

Let $U = ((T_2, T_{1,2}, \ldots, T_{1,p}), T_{1,1})$. Then $C(U) = C(T)$ (Lemma 10), $|U| = |T|$ and $s(U) = 2 = s(T)$. The assumption gives

$$|T_{1,1}| = g(T_1) > |T| - |T_1| = |T_2|. \tag{4}$$

Since $T_{1,1}$ is a subtree of T_1

$$|T_{1,1}| < |T_1|. \tag{5}$$

There are to subcases: $g(U) = |T_{1,1}|$ and $g(U) = |(T_2, T_{1,2}, \ldots, T_{1,p})|$. In subcase $g(U) = |T_{1,1}|$ we have

$$g(U) = |T_{1,1}| \underset{(5)}{<} |T_1| = g(T).$$

For subcase $g(U) = |(T_2, T_{1,2}, \ldots, T_{1,p})|$ we have

$$g(U) = |(T_2, T_{1,2}, \ldots, T_{1,p})| \underset{(4)}{<} |(T_{1,1}, T_{1,2}, \ldots, T_{1,p})| = |T_1| = g(T).$$

Therefore $U < T$ and $C(U) = C(T)$. This again contradicts the choice of T as a minimal counterexample.

Claims 1–5 show that $s(S) = 2 \leqslant s(T)$, $g(S) = |S_1| \leqslant |T_1| = g(T)$ and

$$|S| = |S_1| + g(S_1) \leqslant |T_1| + g(T_1) \leqslant |T|.$$

This contradicts the assumption that there is no $S \in \mathscr{F}_{q,r+1}$ with $S \leqslant T$. □

Proof of the second part of Theorem 1. Let $T \in \mathscr{T}_{q,r}$. By Lemma 12 there is a tree $S \in \mathscr{F}_{q,r}$ with $(|S|, g(S), s(S)) \leqslant (|T|, g(T), s(T))$. By Lemma 8 there is $(n, a, b) \in m_{q,r}$, with $(n, a, b) \leqslant (|S|, g(S), s(S))$. The relation $\leqslant$ is transitive so we have $(n, a, b) \leqslant (|T|, g(T), s(T))$. □

We have seen that for $q, r \in \mathbb{N}$, $q \geqslant 2$ and $r \geqslant 1$ we can choose a tree T with search complexity $r + 1$ and minimal numbers of leafs (i.e. $\mu_{q,r+1}$ leafs) from $\mathscr{F}_{q,r+1}$. By definition there must be a tree $S \in \mathscr{F}_{q,r}$ with $T \in Y(S)$ or $T \in N(S)$. However usually the tree S has more than $\mu_{q,r}$ leafs. This is the reason why there is no trivial linear recurrence equation for the series $(\mu_{q,r})_{r \in \mathbb{N}}$.

4. Asymptotic behavior

In this section we will prove Theorem 4.

Lemma 13. *For every* $q,m,n\in\mathbb{N}$, $q\geqslant 2$, $m,n\geqslant 1$

$$\mu_{q,m+n}\leqslant \mu_{q,m}\cdot\mu_{q,n}.$$

Proof. Let S_1,S_2 be q-trees with $C(S_1)=m$, $|S_1|=\mu_{q,m}$, $C(S_2)=n$ and $|S_2|=\mu_{q,n}$ (cf. Corollary 3).

Construct a q-tree T as follows: For every leaf x of S_1 remove x and connect the parent of x with the root of a copy of S_2. T has $|S_1|\cdot|S_2|$ leaves. We will now show that $C(T)=m+n$:

Let A_i be an optimal search algorithm for S_i, $i=1,2$. A possible search algorithm A for the tree T is:

Use A_1 on the copy of S_1 in T
Use A_2 on the remaining copy of S_2
This algorithm uses $L(A)=C(S_1)+C(S_2)$ questions. Therefore

$$C(T)\leqslant L(A)=C(S_1)+C(S_2)=m+n. \tag{6}$$

Let B_i be an optimal oracle for S_i, $i=1,2$. W.l.o.g. we assume that B_1 has the following property: Whenever the search algorithm asks a leaf of S_1 then the answer from B_1 is "No". If B_1 had answered "Yes" instead then the search algorithm would have found the leaf. The answer can therefore be "No" without reducing the number of needed questions.

The oracle B for T is defined in the following way:

Use B_1 for the first $C(S_1)$ questions with the following extension:
If the node asked for is not in S_1 **then** answer "No".
Use B_2 on the remaining copy of S_2.

Let A be an arbitrary search algorithm. We may assume w.l.o.g. that the first m questions are nodes in S_1. If the algorithm asks for a node x in a subtree S_2 then the answer is "No". The same answer is given if the root y of S_2 which is a leaf of S_1 is asked. Therefore $L(A,B)$ cannot be greater if the question T_x is replayed by T_y.

The oracle B answers the first m questions just like the oracle B_1 would. At least m questions are necessary until A has reduced the tree S_1 to a leaf since $C(S_1)=m$ and B_1 is optimal. This implies that after m questions A has reduced the tree S to a tree which has at least one complete copy of S_2 as a subtree and so at least $n=C(S_2)$ more questions are necessary. This yields $L(A,B)\geqslant m+n$. Since A was arbitrary we get

$$C(T)\geqslant m+n. \tag{7}$$

(6) and (7) show that $C(T)=m+n$.

Since $C(T)=m+n$, the smallest q-tree with search complexity at least $m+n$ can have at most $|T|$ leaves. Therefore

$$\mu_{q,m+n}\leqslant |T|=\mu_{q,m}\cdot\mu_{q,n}. \qquad \square$$

A nonnegative function $f : \mathbb{N} \setminus \{0\} \to \mathbb{R}_+$ is called *subadditive* if

$$f(m+n) \leqslant f(m) + f(n)$$

for all $m, n \in \mathbb{N} \setminus \{0\}$.

The Fundamental Lemma for subadditive functions is: For every subadditive function f the series $(f(n)/n)_{n\in\mathbb{N}}$ converges and

$$\lim_{n\to\infty} \frac{f(n)}{n} = \inf_{n\geqslant 1} \frac{f(n)}{n}.$$

Using this lemma we get a proof for the convergence of $\left(\sqrt[r]{\mu_{q,r}}\right)_{r\in\mathbb{N}}$.

Proof of the first part of Theorem 4. Let $f(r) = \log(\mu_{q,r})$. Lemma 13 gives that f is subadditive. Therefore the limit

$$\lim_{r\to\infty} \frac{f(r)}{r} = \inf_{r\geqslant 1} \frac{f(r)}{r}$$

exists.

$x \mapsto \mathrm{e}^x$ is continuous and strong isotone. Therefore we have

$$\inf_{r\geqslant 1} \sqrt[r]{\mu_{q,r}} = \inf_{r\geqslant 1} \mathrm{e}^{f(r)/r} = \mathrm{e}^{\inf_{r\geqslant 1} f(r)/r} = \mathrm{e}^{\lim_{r\to\infty} f(r)/r} = \lim_{r\to\infty} \mathrm{e}^{f(r)/r} = \lim_{r\to\infty} \sqrt[r]{\mu_{q,r}}$$

and the limit exists. □

We will now derive the upper bound of Theorem 4.

Lemma 14. *For all $q, r' \in \mathbb{N}$, $q \geqslant 2$ and $r := qr' + 1$ there is a tree $T_r \in \mathscr{F}_{q,r}$ with*

$$\begin{pmatrix} |T_r| \\ g(T_r) \end{pmatrix} = \begin{pmatrix} q & 1 \\ q-1 & 1 \end{pmatrix}^{r'} \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix}. \tag{8}$$

Proof. $r' = 0$: We have $r = 1$. The tree in $\mathscr{F}_{q,1}$ fulfills the claim.

$r' \to r' + 1$: Let T be an arbitrary Fibonacci tree and let $T' \in Y(T)$ and $T'' \in N(T)$. The following equalities are easy to verify:

$$\begin{pmatrix} |T'| \\ g(T') \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} |T| \\ g(T) \end{pmatrix} \tag{9}$$

and

$$\begin{pmatrix} |T''| \\ g(T'') \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} |T| \\ g(T) \end{pmatrix}. \tag{10}$$

By induction we have a tree $T_r \in \mathscr{F}_{q,r}$ that fulfills (8). Let

$$T_{r+q} = Y(\underbrace{N(N(\ldots N(}_{q-2\text{times}} Y(T_r))\ldots))).$$

T_{r+q} is in $\mathscr{F}_{q,r+q}$ since the operator N is never executed on a tree S with $s(S) = q$. We get

$$\begin{aligned}
\begin{pmatrix} |T_{r+q}| \\ g(T_{r+q}) \end{pmatrix} &\underset{(9),(10)}{=} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{q-2} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} |T_r| \\ g(T_r) \end{pmatrix} \\
&= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & q-2 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} |T_r| \\ g(T_r) \end{pmatrix} \\
&= \begin{pmatrix} q & 1 \\ q-1 & 1 \end{pmatrix} \cdot \begin{pmatrix} |T_r| \\ g(T_r) \end{pmatrix} \\
&\underset{\text{induction}}{=} \begin{pmatrix} q & 1 \\ q-1 & 1 \end{pmatrix}^{r'+1} \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix}. \qquad \square
\end{aligned}$$

Lemma 15. *Let $q \in \mathbb{N}$, $q \geqslant 2$. The matrix*

$$A := \begin{pmatrix} q & 1 \\ q-1 & 1 \end{pmatrix}$$

has two distinct real eigenvalues $\alpha = (q + 1 + \sqrt{q^2+2q-3})/2$ and $\beta = (q+1-\sqrt{q^2+2q-3})/2$. In particular $0 < \beta < \alpha$.

$$v_\alpha = \begin{pmatrix} 1 \\ \alpha - q \end{pmatrix} \quad \text{and} \quad v_\beta = \begin{pmatrix} 1 \\ \beta - q \end{pmatrix}$$

are the eigenvectors to the respective eigenvalues. The vector $w := \binom{2}{1}$ can be written as $w = c_1 v_\alpha + c_2 v_\beta$ with $c_1 = (1 + 2q - 2\beta)/\alpha - \beta > 0$ and $c_2 = (-1 - 2q + 2\alpha)/\alpha - \beta$.

Proof. We have $\sqrt{q^2+2q-3} < \sqrt{q^2+2q+1} = q+1$. This shows that $0 < \beta < \alpha$.

The vectors v_α and v_β fulfill $Av_\alpha = \alpha v_\alpha$ and $Av_\beta = \beta v_\beta$.

The claims $w = c_1 v_\alpha + c_2 v_\beta$ and $c_1 > 0$ can be verified easily. $\square$

Proof of the second part of Theorem 4. We will show the claimed inequality for the subsequence $({}^{qr'+1}\sqrt{\mu_{q,qr'+1}})_{r' \in \mathbb{N}}$. This is sufficient since the subsequence has the same limit as the sequence $({}^{r}\sqrt{\mu_{q,r}})_{r \in \mathbb{N}}$ and we have already proved the convergence of this sequence.

Let $r' \in \mathbb{N}$, $r = qr' + 1$ and T_r be the tree from Lemma 14. From Lemma 11 we get $C(T_r) = r$. Corollary 3 therefore implies $\mu_{q,r} \leqslant |T_r|$.

In the following we use the notation from Lemma 15.

$$\begin{pmatrix} |T_{qr'+1}| \\ g(T_{qr'+1}) \end{pmatrix} \underset{\text{Lemma } 14}{=} A^{r'} \cdot w = A^{r'} \cdot (c_1 v_\alpha + c_2 v_\beta) = c_1 \alpha^{r'} v_\alpha + c_2 \beta^{r'} v_\beta.$$

We have

$$\lim_{r' \to \infty} \frac{qr'}{qr' + 1} = 1, \tag{11}$$

and since $0 < \beta < \alpha$ and $c_1 > 0$ we get

$$\lim_{r' \to \infty} \frac{\log(c_1 \alpha^{r'} + c_2 \beta^{r'})}{\log(\alpha^{r'})} = 1. \tag{12}$$

Therefore

$$\begin{aligned}
\lim_{r' \to \infty} \sqrt[qr'+1]{|T_{qr'+1}|} &= \lim_{r' \to \infty} \sqrt[qr'+1]{c_1 \alpha^{r'} + c_2 \beta^{r'}} \\
&= \lim_{r' \to \infty} \exp\left(\frac{1}{qr' + 1} \log(c_1 \alpha^{r'} + c_2 \beta^{r'}) \right) \\
&= \exp\left(\lim_{r' \to \infty} \frac{1}{qr' + 1} \log(c_1 \alpha^{r'} + c_2 \beta^{r'}) \right) \\
&= \exp\left(\frac{\log(\alpha)}{q} \lim_{r' \to \infty} \frac{qr'}{qr' + 1} \frac{\log(c_1 \alpha^{r'} + c_2 \beta^{r'})}{\log(\alpha^{r'})} \right) \\
&\underset{(11),(12)}{=} \exp\left(\frac{\log(\alpha)}{q} \right) \\
&= \sqrt[q]{\alpha}.
\end{aligned}$$

From $\mu_{q,qr'+1} \leqslant |T_{qr'+1}|$ for all $r' \in \mathbb{N}$ we can conclude

$$\lim_{r' \to \infty} \sqrt[qr'+1]{\mu_{q,qr'+1}} \leqslant \sqrt[q]{\alpha} = \sqrt{\frac{q + 1 + \sqrt{q^2 + 2q - 3}}{2}}. \qquad \square$$

5. A note about the conjecture

The upper bound is derived with the trees from Lemma 14. Of course every other sequence of trees in $\mathscr{F}_{q,r}$ also gives an upper bound for $\sqrt[r]{\mu_{q,r}}$. We know from Theorem 1 and Corollary 3 that there is a sequence of trees $S_r \in \mathscr{F}_{q,r}$ for which the derived bound is sharp.

A sequence one might want to test is $T'_1 \in \mathscr{F}_{q,1}$ and

$$T'_{(q-1)(r'+1)+1} \in Y(\underbrace{N(N(\ldots N(}_{q-2\text{times}} T'_{(q-1)r'+1})\ldots))).$$

The bound which can be derived by this sequence is

$$\sqrt{\frac{q-1+\sqrt{q^2-2q+5}}{2}}.$$

However it can be shown (cf. [4, Theorem 4.17]) that for all $q \in \mathbb{N}$ with $q \geqslant 3$ we have

$$\sqrt{\frac{q-1+\sqrt{q^2-2q+5}}{2}} > \sqrt{\frac{q+1+\sqrt{q^2+2q-3}}{2}}.$$

This seems paradoxical since all inner nodes of the trees $T'_{(q-1)r'+1}$ have exactly q children, whereas the trees $T_{qr'+1}$ have many inner nodes with only two children.

Acknowledgements

I want to thank Martin Aigner for his support during my research on this topic. I further thank Arno Wagner and Carsten Damm for proofreading.

References

[1] M. Aigner, Combinatorial Search, Teubner-Wiley, Leipzig/New York, 1988.
[2] D. Du, F.K. Hwang, Combinatorial Group Testing and its Applications, World Scientific, Singapore, 1993.
[3] R.L. Rivest, The game of "N" questions on a tree, Discrete Math. 17 (1977) 181–186.
[4] F. Recker, Kombinatorische Suche in Bäumen, Ph.D. Thesis, Department of Mathematics, Freie Universität Berlin, 1997.